09/332919

WHAT IS CLAIMED IS:

1. A graphics system comprising:

a control unit configured to receive compressed 3D geometry data, wherein said compressed 3D geometry data comprises a plurality of blocks; and

a plurality of decompress pipelines, wherein said control unit is configured to selectively route said blocks to one or more of said decompress pipelines, wherein each block comprises compressed vertex information, and wherein said plurality of decompress pipelines are configured to decompress said blocks into a plurality of vertices.

2. The graphics system as recited in claim 1, wherein said decompress pipelines are configured to cache said blocks of compressed vertex information to a memory.

3. The graphics system as recited in claim 2, wherein said decompress pipelines are configured to retrieve said cached blocks of compressed vertex information from said memory in a just-in-time manner.

4. The graphics system as recited in claim 1, wherein said control unit is configured to route said blocks to said one or more of said decompress pipelines in a just-in-time manner.

5. The graphics system as recited in claim 1, wherein each decompress pipeline is coupled to one or more transform units configured to transform said vertices from a first reference coordinate system to a second reference coordinate system.

6. The graphics system as recited in claim 5, wherein each transform unit is coupled to a lighting unit configured to perform one or more vertex processes on said transformed vertices, and wherein each lighting unit is coupled to one or more setup units, wherein said setup units each comprise a transformed vertex memory, wherein each setup unit

is configured to store selected processed vertices into said transformed vertex memory, and wherein each setup unit is configured to reuse said selected processed vertices stored in said transformed vertex memory to form said geometric primitives.

7. The graphics system as recited in claim 1, wherein each decompress pipeline is coupled to one or more set up units configured to transform said vertices and assemble geometric primitives from said transformed vertices, wherein each setup unit comprises a transformed vertex memory, wherein each setup units is configured to store vertices into said transformed vertex memory, and wherein said setup units are configured to reuse said transformed vertices stored in said transformed vertex memory to form geometric primitives.

8. The graphics system as recited in claim 7, wherein said transform units are configured to receive and transform vertices independently of the geometric primitives to which said vertices belong.

9. The graphics system as recited in claim 7, wherein said transform units are configured to receive and transform vertices independently of the geometric primitives to which said vertices belong, wherein there are no state changes between vertices.

10. The graphics system as recited in claim 7, wherein said lighting units are configured to perform lighting calculations independently of the geometric primitives to which said vertices belong.

11. The graphics system as recited in claim 7, wherein said vertices are subjected to one or more vertex processes before being stored into said transformed vertex memory and before being used to form geometric primitives.

12. The graphics system as recited in claim 7, wherein said transformed vertices stored in said transformed vertex memory comprise xyz position information, color information, and transparency information.

13. The graphics system as recited in claim 12, wherein said transformed vertices further comprise additional per-graphics primitive attributes.

14. A method for decompressing and rendering compressed 3D geometry data comprising:

receiving said compressed 3D geometry data, wherein said compressed 3D geometry data comprises a plurality of blocks;

detecting control information within said compressed 3D geometry data;

routing said blocks to one or more decompressors according to said control information, wherein said decompressors are configured to decompress each said block into a plurality of vertices and a plurality of corresponding mesh buffer references;

transforming said vertices from a first coordinate reference frame to a second coordinate reference frame;

performing additional vertex processes on said transformed vertices; and

assembling geometric primitives using said processed vertices.

15. The method as recited in claim 14, wherein said transforming is performed on multiple vertices in parallel.

16. The method as recited in claim 14, wherein said performingadditional vertex processes is performed on multiple vertices in parallel.

17. The method as recited in claim 14, further comprising maintaining vertex reuse information corresponding to said vertices throughout said decompressing, said transforming, said performing additional vertex processes, wherein said vertex reuse information is utilized during said assembling.

18. The method as recited in claim 14, wherein said vertex reuse information comprises mesh buffer references accompanying said vertices in said compressed 3D geometry

data.

19. The method as recited in claim 14, wherein said transforming and said performing additional vertex processes are performed on a per-vertex basis without regard to which geometric primitives the vertices belong.

20. The method as recited in claim 19, wherein there are no state changes between vertices.

Sub
a1

21. A computer system comprising:

a central processing unit (CPU);

a memory; and

a graphics system, wherein said CPU, said memory and said graphics system are coupled by one or more buses, and wherein said graphics system comprises:

a control unit configured to receive compressed 3D geometry data, wherein said compressed 3D geometry data comprises a plurality of blocks; and

a plurality of decompress pipelines, wherein said control unit is configured to selectively route said blocks to one or more of said decompress pipelines, wherein each block comprises compressed vertex information, and wherein said plurality of decompress pipelines are configured to decompress said blocks into a plurality of vertices.

22. The computer system as recited in ~~claim~~ 21, wherein said control unit is configured to detect control information embedded within said compressed 3D geometry data, and wherein said control unit is configured to ~~route said~~ compressed 3D geometry data to one or more of said decompress pipelines according to said embedded control information.

Add
a2